kvm-unit-tests

Andrew Jones <ajones@ventanamicro.com>
About me

- Work on RISC-V Linux, KVM, OpenSBI and QEMU at Ventana Micro Systems
- Participate in RVI working groups and RISE
- Previously worked on AArch64 Virtualization at Red Hat
- Ported kvm-unit-tests to arm and aarch64, and continue to maintain those ports
- Initial port of kvm-unit-tests to powerpc64
- And now bringing RISC-V to kvm-unit-tests
Outline

- kvm-unit-tests
- kvm-unit-tests and RISC-V
- kvm-unit-tests and you
kvm-unit-tests

- As old as KVM, originally created to test qemu-kvm
- Can now test other KVM userspaces, other hypervisors, and hardware
- Supports arm, aarch64, ppc64, s390x, i386, x86_64, riscv32, riscv64
- Unit tests are tiny guest kernels
  - The tests and any additional threads of execution are booted
    - DT and limited ACPI support
    - EFI-supporting bootloaders (aarch64, x86_64, and (wip) riscv64)
    - Linux boot requirements for direct boot (arm and riscv)
  - The tests typically run in the same privilege mode as a kernel, allowing...
    - execution of privileged instructions
    - manipulation of page tables and exception vectors
    - ...
**kvm-unit-tests**

- Despite running in kernel-mode, tests look like typical C applications
  - `main(argc, argv)`
  - Environment variables may be provided
  - Some favorite libc functions; `printf`, `malloc`, `strcmp`, `strtol`, `assert`, ...

```c
int main(int argc, char **argv)
{
    ...
    expected = strtol(getenv("MVENDORID"), NULL, 0);

    ret = sbi_ecall(SBI_EXT_BASE, SBI_EXT_BASE_GET_MVENDORID, 0, 0, 0, 0, 0, 0, 0);
    report(!ret.error, "mvendorid: no error");
    report(ret.value == expected, "mvendorid");

    ...
}
```
kvm-unit-tests

- Load the unit test however necessary, e.g. QEMU’s -kernel parameter

- Also has a simple test harness
  - run all tests or
  - run all tests in a group

- Test harness is written in Bash

- Can build standalone tests
  - A standalone test is a single Bash script with the compiled unit test embedded (base64)
  - The idea is to be able to email reproducers
kvm-unit-tests and RISC-V

- SBI verification framework
  - Functional testing for SBI implementations, e.g. OpenSBI and RustSBI, KVM

- CPU validation
  - For example, memory model litmus tests (Arm does this)
  - QEMU

- Microbenchmarks
  - Test instruction sequences in isolation

- KVM and KVM userspace tests
kvm-unit-tests and you

- Tests are easy to write, quick&dirty tests are even easier to write, but...
- Ideally we’d get tests posted for merge
  - Allows the framework to evolve
  - Other tests provide examples for new tests
  - Is “one and done” ever true anyway?
- Post here: kvm-riscv@lists.infradead.org
- Really doing think the test belongs with the others?
  - unittests.cfg
  - Considering another branch to collect tests like these
References

- Main kvm-unit-tests repo
  https://gitlab.com/kvm-unit-tests/kvm-unit-tests

- Branch with ADUE test
  https://gitlab.com/jones-drew/kvm-unit-tests/-/tree/riscv/x-tests

- [Outdated] write-up about kvm-unit-tests
  https://www.linux-kvm.org/page/KVM-unit-tests
Thank You